24.1 4GHz+ Low-Latency Fixed-Point and Binary Floating-Point Execution Units for the POWER6 Processor

B. Curran[1], B. McCredie[2], L. Sigal[3], E. Schwarz[1], B. Fleischer[3], Y-H. Chan[1], D. Webber[1], M. Vaden[2], A. Goyal[2]

[1]IBM, Poughkeepsie, NY
[2]IBM, Austin, TX
[3]IBM, Yorktown Heights, NY

Traditional high-frequency circuit designs have relied on deep pipelines and/or aggressive dynamic circuits. New low-latency static-circuit fixed-point (FXU) and binary floating-point (BFU) units are described that naturally achieve high frequency without resorting to super deep pipelines or power-hungry dynamic circuits. These POWER6 units are designed and fabricated in a 65nm SOI process technology and are currently operating in excess of 4GHz. Driving frequency for sake of frequency can simply result in higher power with little or no performance gain. Non-critical circuits in these units use high-V_T devices to minimize leakage. Power simulations at 1.1V, 4GHz, and 100% utilization show POWER6 FXU and BFU dataflow consume 160mW and 310mW, respectively. When not utilized, the execution dataflow latches are clock gated to achieve zero active power. Higher frequency, lower power, lower execution latency and higher performance relative to the prior design [1] are simultaneously achieved.

The one pipeline stage, 13 FO4 fixed-point engine allows back-to-back execution of data dependent add, subtract, compare, rotate, shift, and logical instructions. (13 FO4 is equal to the delay of 13 inverter stages with fanout of four). 64b results can be forwarded back to either operand of the same FXU or different FXU in superscalar implementations. A previous FXU design had 2-cycle, dependent execution latency of 44 FO4 [1].

Traditionally for subtract and compare operations one of the operands is complemented via an XOR circuit just prior to the adder. The extra delay incurred by the complementing has been eliminated by early detection of dependent subtract or compare operations and early complement of the data within the adder, rotator, and logical sub-units (Fig. 24.1.1). The complementing occurs in non-cycle-limiting paths approximately 1/2 cycle earlier than in prior FXUs. In addition, the rotator is modified from previous design [1] to eliminate the word-replicator MUX used for 32b rotate instructions. Word-replicate function is achieved by computing unique selects for each byte within the rotator. Selection of the adder, rotator, and logical sub-unit results is another delay overhead incurred in traditional FXU designs. This delay is reduced by a distributed MUX circuit. Early decode of the opcode determines which subunit result will be selected. Unselected subunits drive their result outputs to logical one state; this drive-to-one function also occurs in non-cycle-limiting paths. The final selector function, which is part of the adder, simplifies to a fast NAND3 circuit.

The FXU operands have many sources: register file, recent FXU results awaiting register-file writeback, hot fixed-point results from prior cycle of one or more FXUs, and load data. A clocked MUX operand latch accommodates all these sources with minimal multiplexer delay overhead (Fig. 24.1.2). All multiplexer inputs as well as the latch output are static. This latch is scannable and able to initiate at-speed ac test via the c2 clock. The FXU incorporates several logic and circuit innovations to achieve a 13 FO4 cycle with a back-to-back execution capability.

The BFU has a 7 cycle, 13 FO4 per cycle fused multiply-add pipeline (Fig. 24.1.3). New instructions can be started every cycle, data-dependent instructions can be initiated after just 6 cycles in most cases. The previous BFU design had an effective length of 132 FO4 [1]. There are also no pipeline stalls even in the case of denormalized input or massive cancellation.

The fused multiply-add dataflow supports up to 64b fixed-point or floating-point operands into the multiplier and a 53b addend. The input operands can be sourced from register file or recent BFU results that may be unrounded and only partially normalized [2]. The radix-4 multiplier creates 33 partial products from the 64b operands. Leading-one correction terms are added to the partial product array to correct for denormalized input. These terms are directly concatenated to the least significant two partial products. An additional partial product is added late to perform a rounding correction. The partial product array is reduced using 4:2 and 3:2 counters. In parallel, the shift amount is calculated for the addend and it is aligned to the product. In cycle 3, the addend is combined with two partial products using a 3:2 counter. The 128b adder is an end-around-carry parallel prefix 2 static adder distributed across three cycles. In cycle 3, the 1b propagate and generate are computed and latched. In cycle 4, the adder produces a conditional sum on a 32b basis and latches both sum and sum plus 1. In cycle 5, the adder gates the correct sum for each 32b group and conditionally complements the result. Normalization is split between cycle 5 and cycle 6 and rounding is split across cycles 6 and 7.

To achieve a 6 cycle feedback, an unrounded result and rounding correction term are forwarded to dependent instructions. The rounding correction term is incorporated into the multiplier. A dependency of the addend on the prior result can be corrected by bypassing the result exponent after the normalizer but bypassing the significand after rounding into the latch feeding cycle 2. In addition, the wire delays of a traditional bit stack are excessive. To minimize these delays an "O" shape floorplan is created whereby the BFU computation flows down the right stack to the adder and then flows up the left stack back to the operand registers (Fig. 24.1.4). The adder is the optimal cross-over point since only the group generates are timing critical and require fast low-resistance horizontal wires. One further enhancement technique in shift left one (SL1) correction is necessary. Since most leading-zero-anticipator (LZA) designs can be off by 1; a traditional SL1 circuit is needed to repower a high-order data bit to select all bits of a very wide MUX. The SL1 delay overhead is avoided by forwarding an unrounded non-SL1-corrected 1b-wider result. A masking function is designed for the least significant bits (bit 24 for single and bit 53 for double precision.)

To minimize latch delay overhead, the BFU dataflow makes extensive use of scannable pulsed latches that can operate in two modes (Fig. 24.1.5). In high-performance mode, the pulsed c2_chop clock is activated to write data directly into the L2 latch. In robust (safety) mode, the c1 and c2 clocks are activated such that the latch operates in a traditional master-slave. This mode is primarily used during burn-in to eliminate potential pulse width and latch writability problems at high V_{dd} and high temperature. The POWER6 BFU incorporates many microarchitecture, logic, circuit, latch, and integration techniques to achieve an effective 6-cycle 13-FO4 pipeline.

References:

[1] J.M. Tendler, J.S. Dodson, J.S. Fields, Jr., H. Le, B. Sinharoy, "Power4 System Microarchitecture," *IBM Journal of Research and Development*, vol. 46, no. 1, pp. 5-25, Jan., 2002.
[2] E.M. Schwarz, M. Schmookler, S.D. Trong, "FPU Implementations with Denormalized Numbers," *IEEE Transactions on Computers*, vol. 54, no. 7, pp. 825-836, Jul., 2005.

 1-4244-0079-1/06/$20.00

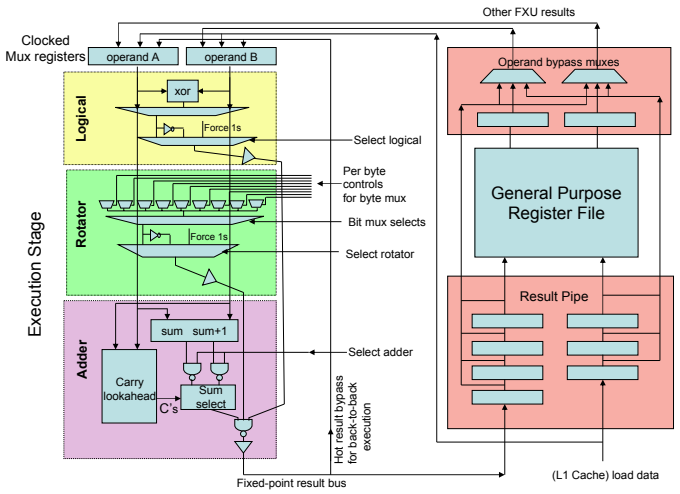

Figure 24.1.1: Power6 fixed-point execution unit.

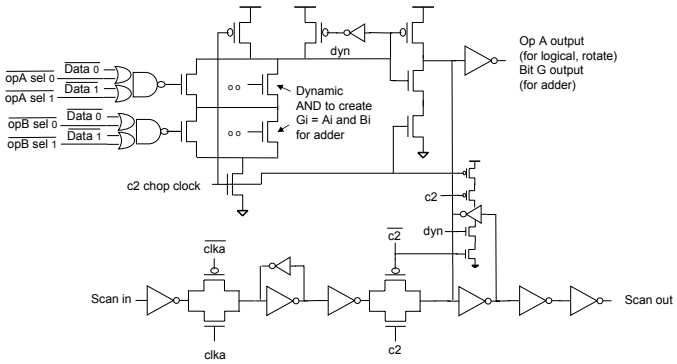

Figure 24.1.2: Clocked-MUX latch for fixed-point operand register.

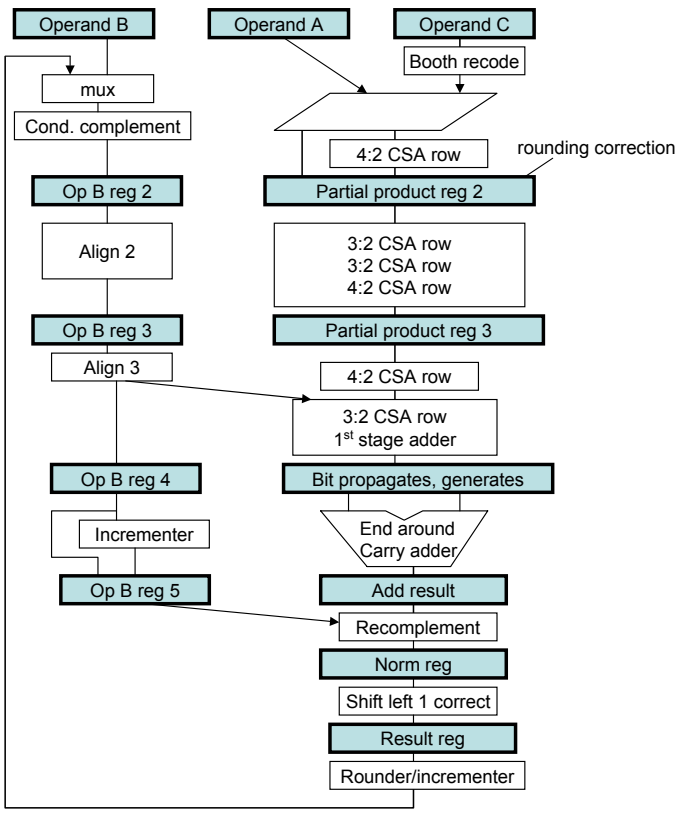

Figure 24.1.3: Power6 binary floating-point fraction dataflow.

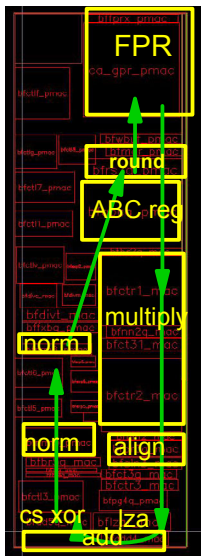

Figure 24.1.4: Binary floating-point "0" floorplan.

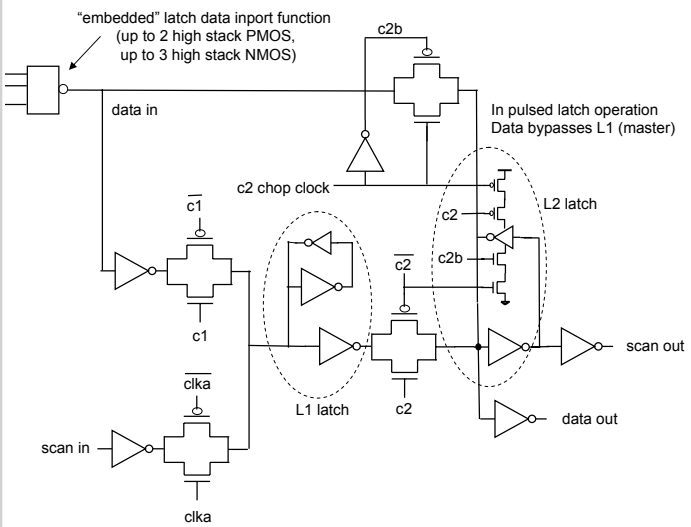

Figure 24.1.5: High-performance pulsed latch.